club commodore

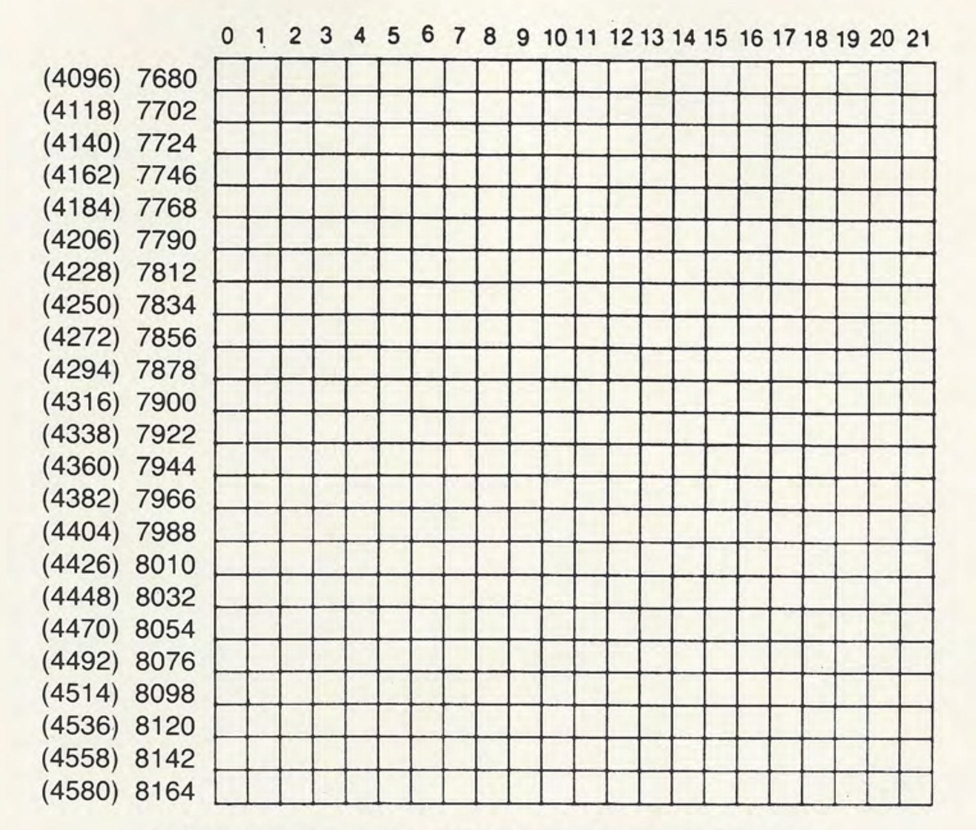
Boletín informativo para los usuarios de microordenadores VIC y CBM



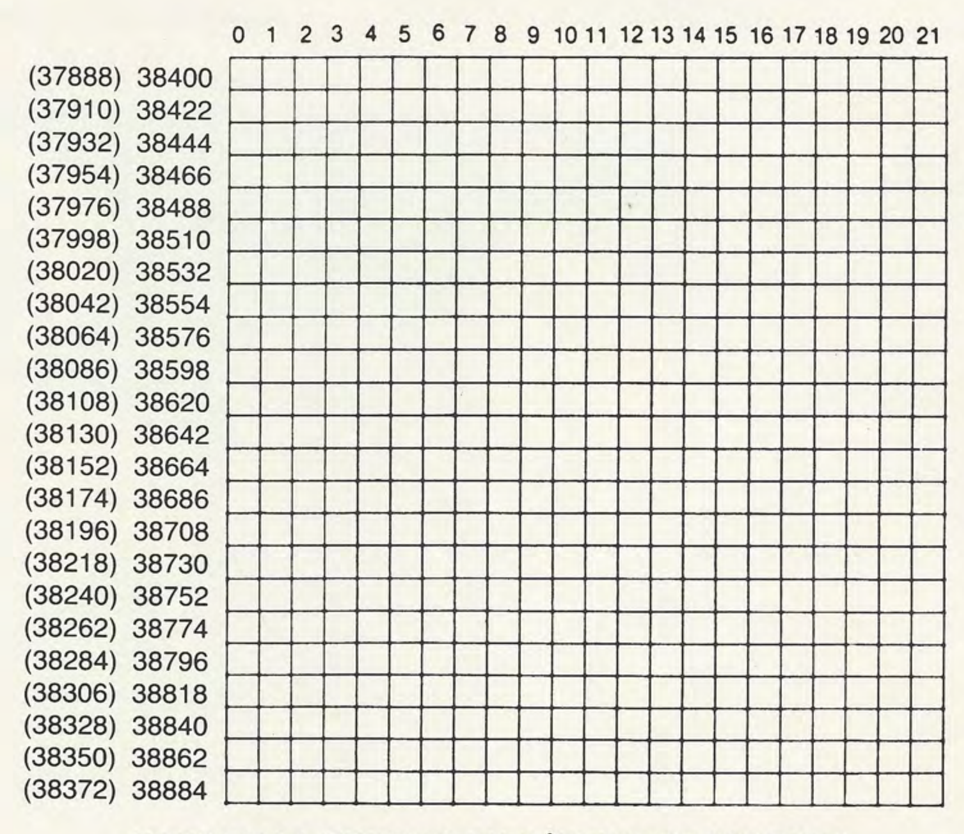
N°/14 diciembre 1983

MAPAS DE LAS MEMORIAS DE PANTALLA Y DE COLOR-VIC-20

(MÁS DE 8 K DE RAM)



CÓDIGOS DE LOS CARACTERES EN PANTALLA



MAPA DE LA MEMORIA DE CÓDIGOS DE COLORES

Depósito Legal Sep. B. 2133-1958 - Altés, s.l., Caballero 87, Barcelona

EDITORIAL

los programas para los ordenadores personales

Para los ordenadores personales suele haber multitud de programas. De hecho, la cantidad y calidad del software disponible para determinado equipo suele ser decisivo a la hora de vender esta máquina, y esta variedad crea en el usuario cierta confusión.

Veremos a continuación las cuatro grandes categorías en las que se pueden clasificar todos los programas existentes.

En primer lugar, existen los programas "a medida", es decir, que han sido desarrollados por un programador o grupo de programadores siguiendo las especificaciones dictadas por el cliente. El futuro de estos programas no es muy brillante porque su costo suele ser muy elevado debido a la inversión de tiempo por parte de personal altamente especializado. Sólo se pueden justificar en aquellos casos en los que la aplicación se realizará en cantidades masivas, cuando no existe otra solución o el costo de resolver el problema sin la ayuda del ordenador es lo suficientemente alto para que justifique la inversión del desarrollo.

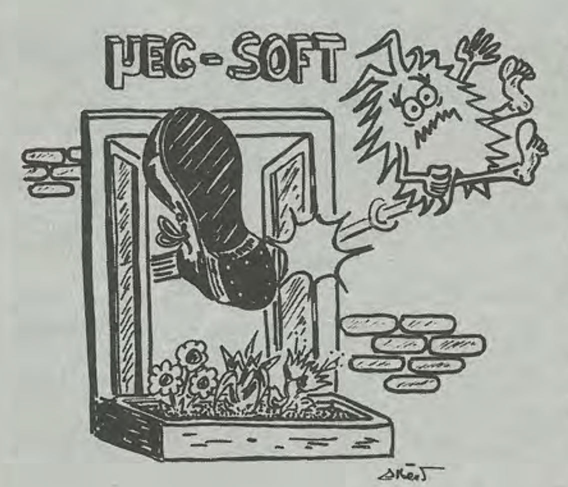
La categoría siguiente cubre todos aquellos programas que no necesitan conocimientos de programación por parte del usuario para utilizarlos. Simplemente, con el programa se adjunta un manual de instrucciones que enseña cómo debe usarse. A estos programas les podemos llamar "programas herramienta" dado que permiten realizar trabajos concretos. Este tipo de software es el que posee un futuro más prometedor debido a que su mercado crece al mismo ritmo que la cantidad de usuarios.

Un tipo de software, que está llamado a revolucionar la utilización de los ordenadores personales, es el que se conoce con el nombre de "generador de programas". Aunque en estos momentos no se conocen muchos ejemplos, se sabe que el truco consiste en que un programa, a través de una serie de preguntas que el usuario debe contestar de forma muy concreta, da como resultado otro programa que es el que realizará el trabajo que requiere la aplicación. En algunos casos el programa generado queda redactado en un lenguaje conocido y es, por tanto, posible que —si el usuario posee conocimientos de programación— se realicen perfeccionamientos que optimicen el funcionamiento de la aplicación. Esto último es importante porque un generador de programas representa una gran ayuda para el programador profesional, al permitirle realizar rápidamente aplicaciones convencionales y —en consecuencia— concentrar su atención en los problemas que requieran un trabajo delicado. Estos programas también pueden ayudar al profesional a demostrar que el ordenador personal que intenta vender a un usuario en potencia puede resolver su problema al realizar sobre la marcha la generación de un programa que cumple las especificaciones que el futuro cliente le va dictando.

La cuarta categoría en la que podríamos clasificar el software disponible para los ordenadores personales es la que contiene aquellos programas desarrollados por el mismo usuario. En un departamento de software profesional se trata de conseguir una productividad lo más alta posible, con lo que ciertos trabajos no son viables. Sin embargo, éste no es el caso del usuario particular que dispone —en principio— de todo el tiempo del mundo. Puede decirse —como mínimo— que esta categoría es la que está llamada a suministrarnos las mayores y más agradables sorpresas en el futuro.

Para terminar recordamos que estamos en Diciembre y queremos aprovechar la ocasión para desear a nuestros lectores, colaboradores y —en general— a todos/as los usuarios de ordenadores personales:

**¡FELICES FIESTAS
Y PRÓSPERO AÑO NUEVO!
BON NADAL I FELIÇ 1984!
ZORIONAK ETA URTE BERRI ON
¡BOAS FESTAS!**



VENTANA CBM

READY (del chip a la base de datos) (2)

por **RAFAEL NAVARRO** (M.E.C.-SOFT)

Tal y como vimos en el capítulo anterior, el microprocesador significa la capacidad de programación, la flexibilidad. Los circuitos integrados de generaciones anteriores (lógica cableada) eran concebidos para desempeñar una tarea concreta y sólo una. Los micros, significaban la posibilidad de

(continúa en la pág. siguiente)

VENTANA CBM

READY (del chip a la base de datos)

(continuación)

disponer de una herramienta capaz de aprender su trabajo (ver figura 1).

Muchos de los trabajos que se realizaban o se realizan mediante lógica cableada, pueden ser llevados a cabo y resulta conveniente hacerlo, por un microprocesador o un sistema basado en microprocesadores. Las aplicaciones en la industria son las que menos elementos periféricos tradicionales (discos, impresoras, etc.) y más código máquina emplean. Por eso hablaremos de ellas en primer lugar.

II. LOS TRANSDUCTORES Y LOS CONTROLADORES

Una de las necesidades más generalizadas en el terreno industrial es la de recoger valores de determinadas magnitudes, procesarlos, exhibir sus valores, organizar un estrépito controlado si dichos valores caen fuera de cierto margen y actuar sobre otras variables del proceso en función del estado del mismo. Algunas variables de entrada pueden ser:

— Temperatura ambiental en laboratorios de investigación farmacéutica.

— Presión atmosférica en centros meteorológicos.

— Pesos en ganadería para control de consumo de piensos, por ejemplo, etc., etc.

— Caudales, densidades, posiciones de válvulas, velocidades, etc... en procesos continuos de producción.

A tal efecto, existe una diversidad de dispositivos cuya misión es transformar tales medidas en magnitudes que el procesador pueda manejar (tensiones eléctricas analógicas, series de impulsos digitales, etc...). Tales aparatos son los llamados transductores.

Una vez realizado el tratamiento de estas magnitudes, el procesador debe poder actuar sobre el proceso bajo control para — por ejemplo — conseguir que el producto que se está fabricando reúna unas condiciones óptimas a pesar de las variaciones de los componentes de entrada. En el microprocesador se obtienen señales de tipo digital que deben utilizarse para controlar válvulas, motores, elementos calefactores, etc... A los dispositivos que realizan la traducción de estos datos a las magnitudes que se manejan en un ambiente industrial, se les llama controladores y algunas de sus funciones son:

— Accionamiento de válvulas; grado de apertura y válvulas todo/nada.

— Puesta en marcha y detención de motores y regulación de su velocidad y sentido de giro.

— Control de la potencia disipada por un calefactor eléctrico para obtener una determinada temperatura.

HARDWARE-CHIP + MICROINSTRUCCIONES

PROGRAMA EN CODIGO MAQUINA
+
DECODIFICADOR DE INSTRUCCIONES

EJECUCION DEL PROGRAMA
Y OBTENCION DE RESULTADOS

Fig. 1 - Nivel más bajo de programación. El comportamiento del chip depende de su lenguaje y del programa en código máquina.

III. EL MICROPROCESADOR EN LA INDUSTRIA

Hasta hace pocos años el microprocesador cumplía — en la industria — el papel de procesador central y único que realizaba todas las operaciones de control de un proceso, dando lugar a problemas de excesivo trabajo y comunicaciones conflictivas (los transductores de un mismo proceso pueden estar a grandes distancias del procesador central y entre sí, y el ambiente industrial suele ser — eléctricamente — ruidoso). Un efecto del abaratamiento de los chips LSI es la incorporación de los microprocesadores a los propios transductores y controladores. En los primeros esta solución permite la calibración automática de los sensores, la elaboración previa de los datos a procesar y el envío de estos datos mediante códigos que reducen los errores de comunicación. En los segundos, además de las mejoras de comunicación, es posible la toma de decisiones finales sobre el terreno, mejorando la flexibilidad del sistema.

En las aplicaciones que vamos a enumerar, el micro suele cumplir una misión muy determinada, conteniendo el sistema en el que se halla integrado, una memoria, normalmente ROM, con un programa o conjunto de programas, que difícilmente serán cambiados (de ahí grabarlo en ROM). En tales sistemas, normalmente, las implicaciones de hardware son superiores a las del software.

Son ya bastantes los fabricantes que han comprendido que gran cantidad de los dispositivos de lógica cableada de baja y media integración (SSI y MSI) e incluso algunos componentes de relojería y mecánicos, pueden ser sustituidos con ventaja por un sistema basado en microprocesadores. Aparte de las ventajas técnicas, existen otros factores no menos importantes tales como la imagen de marca y el distanciamiento de la competencia. De este modo, empieza a ser habitual el oír hablar de lavadoras, neveras, videos y otros electrodomésticos de uso frecuente, con dispositivos y «ventajas» posibles gracias a la incorporación de micros.

Redactamos a continuación y de modo sucinto, las aplicaciones más frecuentes del microprocesador, fuera del entorno del micro-ordenador (ordenador basado en microprocesadores):

INDUSTRIA

CAMPO Y APLICACIÓN

TRANSDUCTORES: La mayoría de los transductores proporcionan una salida eléctrica analógica directamente proporcional a la magnitud que controlan. Para que dicha señal sea

VENTANA CBM

aprovechable en un sistema de microprocesadores, es necesario convertirla previamente en digital. Mediante un interfaz adecuado y el control por microprocesador (para la depuración de la señal), la salida del transductor puede ser recibida en el sistema.

CONTROL NUMÉRICO: Mediante sistemas basados en transductores y microprocesadores, se han construido multitud de robots y sistemas de control de fabricación en cadenas de montaje, sistemas de seguridad, caída de un elemento en la cadena, etc. Si los componentes de una pieza o los recorridos de la cadena varían, basta con cambiar el programa.

GRANJAS Y FABRICACIÓN DE PRODUCTOS ALIMENTICIOS: En cooperativas, para el cálculo y composición automática de piensos, recogiendo las cantidades adecuadas de cada silo e imprimiendo, en algunos casos, incluso los albaranes de entrega del producto. También, en las fábricas de productos alimenticios para el control de recepción de los materiales básicos (leche, cereales, etc.) y contabilidad automática de proveedores.
Para controlar el consumo de piensos mediante balanzas con salida digital.

QUÍMICA: Control de temperatura, densidades, colorimetría, etc., en procesos de producción de compuestos, con variación automática del trabajo de motores, calentadores, agitadores, etc.

MECÁNICA: Control de calidad en el proceso de fabricación mediante sensores ópticos, o de las piezas terminadas.

AUTOMOCIÓN

CAMPO Y APLICACIÓN

FRENADO: Mediante sensores colocados en las ruedas, el sistema envía instrucciones, por medio de servos hidráulicos, inteligentes de frenado antideslizante.

CAMBIOS: Los micros sustituyen con ventajas de todo tipo a los sistemas electrónicos de baja escala de integración que constituyen los controladores de la anterior generación para los controles inteligentes de cambio de marchas y velocidad.

CONSUMOS: Mediante el control por transductores de la temperatura del motor, la velocidad, la presión del aire y la posición del acelerador, se han conseguido economías del orden del 15 al 20 por ciento en combustible.

MANTENIMIENTO: En bancos de prueba y puesta a punto de vehículos.

ANTI-ROBO: Mediante sensores en los puntos débiles (puertas, ventanillas, dispositivos de puesta en marcha, etc.) el sistema envía por radio un mensaje de socorro a la central de seguridad.

ORDENADORES DE A BORDO («ON BOARD COMPUTER»): Son ya populares, por su instalación en algunos turismos del mercado nacional, los ordenadores de a bordo. Se trata de auténticos microordenadores cuyas funciones varían de un modelo a otro, permitiendo consultar multitud de datos (alarmas ante variaciones de temperatura, consumo de combustible, autonomía, control de velocidad, anti-robo, etcétera).

TAXÍMETROS: Además de calcular, según tarifas, el importe del trayecto, estos sistemas permiten listar resultados al final de la jornada.

CONTROL DE TRÁFICO: Para controlar el tráfico mediante semáforos e, incluso, informar acerca de cuáles son las vías más fluidas.

MEDICINA

CAMPO Y APLICACIÓN

MONITORIZACIÓN: A través de una unidad central, se controlan los diferentes micros de cabecera de cada paciente. Así, es posible recibir las magnitudes de cada una de las constantes fisiológicas que se desea controlar y acudir en caso de alarma.

PÁNCREAS ARTIFICIAL: Para el control constante de glucemia en la diabetes mellitus, el páncreas artificial permite conocer las concentraciones de azúcar en la sangre del paciente, emitiendo informes y diagramas (perfiles y curvas de glucemia), avisando ante variaciones alarmantes e inyectando, incluso si así se desea, la insulina necesaria para controlar los niveles de glucosa.

MARCAPASOS: Marcapasos computerizados mucho más seguros.

DIAGNÓSTICO: Sistemas de diagnóstico, basados, bien en la auscultación automatizada por parte del sistema o bien en los datos facilitados directamente por el facultativo.

OTRAS APLICACIONES

CAMPO Y APLICACIÓN

ELECTRODOMÉSTICOS: Lavadoras, lavavajillas, sistemas de calefacción centralizados, sistemas hi-fi, equipos de video, control de consumos en el hogar mediante ordenadores personales, como el VIC o el C64, por ejemplo, etc.

JUEGOS: Videojuegos de aplicación única (con un solo juego) o juegos en cartucho para ordenadores personales.

AGENDAS: Agendas electrónicas con memorización de fecha, hora y asunto.

TRADUCTORAS: Traductoras de idiomas con posibilidad de habla.

EDUCACIÓN: A caballo entre el juego y la enseñanza existen multitud de aplicaciones, como el «speak-and-spell» (habla y deletrea) que permite aprender a pronunciar a los niños de habla inglesa (de momento) o los programas (100) didácticos de la serie COMMODORE.

TRAGAPERRAS: La más cara de las aplicaciones de la microinformática.

Existen por supuesto muchísimas más aplicaciones, ya que éstas dependen únicamente de la imaginación de cada uno, y éstos no son más que algunos ejemplos.
En el próximo capítulo hablaremos (creo) de los sistemas operativos. ■

MEA CULPA

La Redacción de CLUB COMMODORE tiene a no muy bien anunciar que gracias a una gracia (¿vale la redundancia?) de BUG se nos ha colado un error en el número anterior de esta serie. En la columna central de la página 3, en la línea tercera a contar desde el final, donde dice STA 0120 debe decir: STA 0201. Pedimos 3.5 K de perdones.

LENGUAJE FORTH (II)

las pilas (stacks) y la notación RPN

por **RAFAEL PARDO**

Como dijimos en el número anterior, FORTH es un lenguaje muy potente y compacto. Aquí seguiremos examinando el manejo de la pila («stack») y entraremos en el uso y disfrute de la notación RPN (Reverse Polish Notation = Notación Polaca Inversa). ¡¡Que os aproveche!!

LA PILA DE PARÁMETROS (PARAMETER STACK)

La Pila de Parámetros se usa principalmente para pasar valores utilizados en la ejecución de las palabras FORTH. Muchas de estas palabras también envían sus resultados a la pila. Los valores se envían a la pila en alguna de las siguientes operaciones:

A. Todos los valores que se escriben en el teclado se enviarán a la pila.

B. Los valores que se encuentran en un texto fuente se enviarán a la pila.

C. Los valores devueltos por una palabra FORTH como resultados se envían también a la pila.

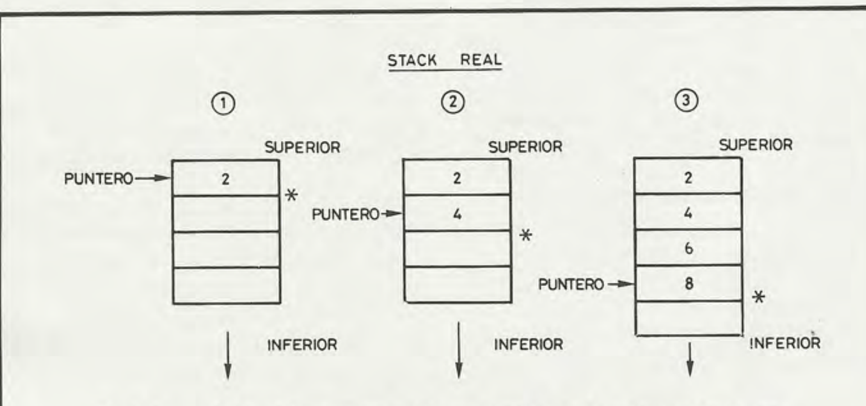
Las operaciones FORTH normalmente destruyen sus argumentos, guardándose sólo los valores en la pila.

Ya que las palabras FORTH necesitan uno o más parámetros de la pila, es necesario asegurarse de que los parámetros están en el orden correcto.

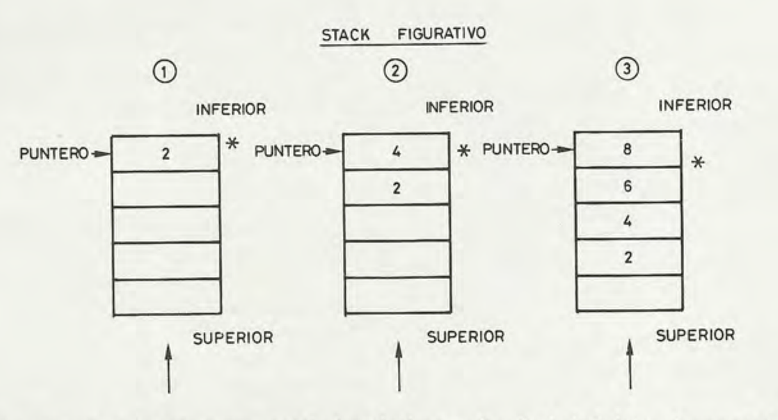
El contenido de la pila se puede manipular por las palabras FORTH de la Tabla 1. El propósito de estas palabras es:

1. Tomar los valores de la pila en el orden correcto para una palabra determinada.
2. Duplicar el valor a tomar de la pila para que ésta no se pierda durante la operación.

La Tabla 2 contiene las palabras que se usan para sacar parámetros de la pila.



La última entrada y/o la primera salida se situarán donde se halla el asterisco.



Dos imágenes de lo que ocurre cuando introducimos en la pila («stack») los números 2, 4, 6 y 8 en este orden.

LA PILA DE RETORNO DE SUBRUTINAS (RETURN STACK)

La otra pila de FORTH se usa para guardar la dirección de la palabra a ejecutar para que el intérprete pueda reconocer la palabra siguiente en orden de ejecución.

La pila de retorno también sirve como un almacén provisional para:

1. Parámetros de bucle.
2. Números que están «de paso» en la pila de parámetros.
3. Punteros para interpretar texto fuente.

Más adelante entraremos en el uso de esta pila.

NOTACIÓN POLACA INVERSA (RPN)

La extrema modularidad de FORTH queda facilitada por el uso que hace de la pila para pasar datos de palabra a palabra. En situaciones complejas, donde los bucles y subrutinas pueden llamarse a varios niveles, la pila se utiliza como un lugar común donde todos los niveles pueden comunicarse. Para aprovechar esta característica, FORTH usa la **Notación Polaca Inversa (RPN)** en vez de las notaciones algebraicas utilizadas comúnmente. Como sugiere la palabra «INVERSA», los operadores se escribirán **después** de los valores que se deban manejar.

En FORTH, los operadores aritméticos siempre trabajan con los valores que en ese momento existen en la pila. Por ejemplo:

126 19 —

colocará el valor 126 en la pila, luego el valor 19, situándose por encima del valor 126 (ver primera parte de esta serie). El signo «—» (menos) restará el segundo valor del primer valor (valor de encima — valor situado inmediatamente debajo) y dejará en la parte superior de la pila **sólo** el resultado.

Considere estos cinco casos:

RPN (FORTH)	**NOTACIÓN NORMAL**
126 19 —	126 — 19
96 6 8 */	(96 * 6) / 8
96 6 8 * /	96 / (6 * 8)
165 45 13 — *	165 * (45 — 13)
102 246 1 2 / + *	102 * (246 + (1 / 2))

Note que el orden de los valores es siempre el mismo en ambos casos y en RPN nunca necesitará un paréntesis. El orden en que se evalúen los valores depende enteramente del orden de los operadores pero nunca es ambiguo. La notación RPN, al eliminar los paréntesis, evita los problemas que suelen producirse si nos olvidamos de cerrar el mismo número de ellos que hemos abierto. Asimismo se evitan los problemas de jerarquía de operadores tan frecuentes en BASIC (por ejemplo).

En la segunda expresión que hemos mostrado anteriormente, aparece "*/" (multiplica-divide).

Ésta es una palabra FORTH que necesita tres valores y que opera con ellos como se ha visto anteriormente, de acuerdo con la notación normal.

Preste atención especial a la tercera expresión mostrada antes. Hay "*" (multiplica) y "/" (divide). Se muestran como operadores separados y que ejecutan la multiplicación y la división respectivamente. Cuando ejecutemos una frase (una serie de palabras dentro de una definición) en FORTH, cada operación se ejecutará en el orden que se vaya encontrando en la línea, de izquierda a derecha. Primero la palabra "*" multiplica el 8 y el 6 (los dos elementos superiores de la pila). Luego la palabra "/" divide la parte superior de la pila (el resultado de la multiplicación anterior, 48) por el elemento que ocupa en este momento la segunda posición en la pila, 96, dando un resultado de 2.

La notación RPN puede resultar algo extraña a primera vista. Pero para algún tipo de aplicación especial, sin embargo, este tipo de notación le simplificará de una manera espectacular el trabajo. Mientras que la notación algebraica normal soporta una notación «descriptiva», la notación RPN soporta «procedimientos». ■

TABLA 1 - **MANIPULACIÓN DE LA PILA (STACK)**

Palabra	**Pila (stack)**	**Efecto**
SWAP	(n1 n2 SWAP n2 n1)	Intercambia los dos valores superiores de la pila.
DUP	(n DUP n n)	Duplica la parte superior de la pila.
—DUP	(n —DUP n n) o (0 —DUP 0)	Duplica el elemento superior sólo si éste no es igual a cero.
DROP	(n DROP)	Saca el valor superior de la pila.
OVER	(n1 n2 OVER n1 n2 n1)	Coloca en lo alto de la pila una copia del segundo elemento de la pila.
ROT	(n1 n2 n3 ROT n2 n3 n1)	Ejecuta una rotación del tercer elemento de la pila hacia arriba.
PICK	(n1 PICK n2)	Sitúa el elemento n1 en la pila.
2SWAP	(d1 d2 2SWAP d2 d1) (n1 n2 n3 n4 2SWAP n3 n4 n1 n2)	Intercambia los dos valores superiores de 32 bits, o los dos pares superiores de 16 bits.
2DUP	(d 2DUP d d) (n1 n2 2DUP n1 n2 n1 n2)	Duplica el valor superior de 32 bits o la pareja superior de 16 bits.
2DROP	(d 2DROP) (n1 n2 2DROP)	Saca el valor superior de 32 bits o la pareja superior de 16 bits.
2OVER	(d1 d2 2OVER d1d2 d1) (n1 n2 n3 n4 2OVER n1 n2 n3 n4 n1 n2)	Copia el segundo valor de 32 bits o la segunda pareja de 26 bits en la parte superior de la pila.
2ROT	(d1 d2 d3 2ROT d2 d3 d1) (n1 n2 n3 n4 n5 n6 2ROT n3 n4 n5 n6 n1 n2)	Ejecuta una rotación del tercer valor de 32 bits hacia arriba (igual con dos de 16 bits).

TABLA 2 - **SALIDA NUMÉRICA DE LA PILA (STACK)**

Orden	**Pila (stack)**	**Efecto**
.	(n .)	Imprime n como un valor con signo seguido de un espacio.
?	(direc. ?)	Imprime el contenido de la dirección como un valor con signo, seguido de un espacio.
D.	(d D.)	Imprime el valor de 32 bits como un valor con signo.
U.	(u U.)	Imprime el valor como un número sin signo.
.R	(n1 n2 .R)	Imprime el valor con signo alineado por la derecha dentro de un campo de n caracteres.
D.R	(d n D.R)	Imprime el número con signo de 32 bits alineado a la derecha en un campo de n caracteres.
H.	(n H.)	Imprime el valor como un número hexadecimal con signo.
B.	(n B.)	Imprime el valor como un número binario con signo.

COLABORACIONES

dibujos matemáticos (R=B×SIN A)

por A. GUIX

A= 100000 B= 30000 T= 3

Se presenta un programa adaptado al COMMODORE 64 y al «Plotter VC 1520» muy simple y que permite obtener una gran variedad de dibujos, tanto simétricos como asimétricos e incluso abiertos. La esencia del algoritmo está en la línea 90. Imagine una pluma montada sobre un brazo que gira respecto al extremo opuesto y cuya longitud (radio) varía en función del ángulo. La pluma trazará curvas y en particular, si R es constante, trazará una circunferencia.

Z es el radio máximo y en el caso del «plotter 1520» puede llegar a valer 240 (240=480/2). Para no ocupar todo el ancho del papel se ha limitado a 200. La variable AN es el valor del ángulo. En la línea 80 se hace variar de 0 a 2×PI radianes en incrementos de un grado (2×PI/360 radianes). Si se desean incrementos de .5 grados, el STEP debe ser 2×PI/720.

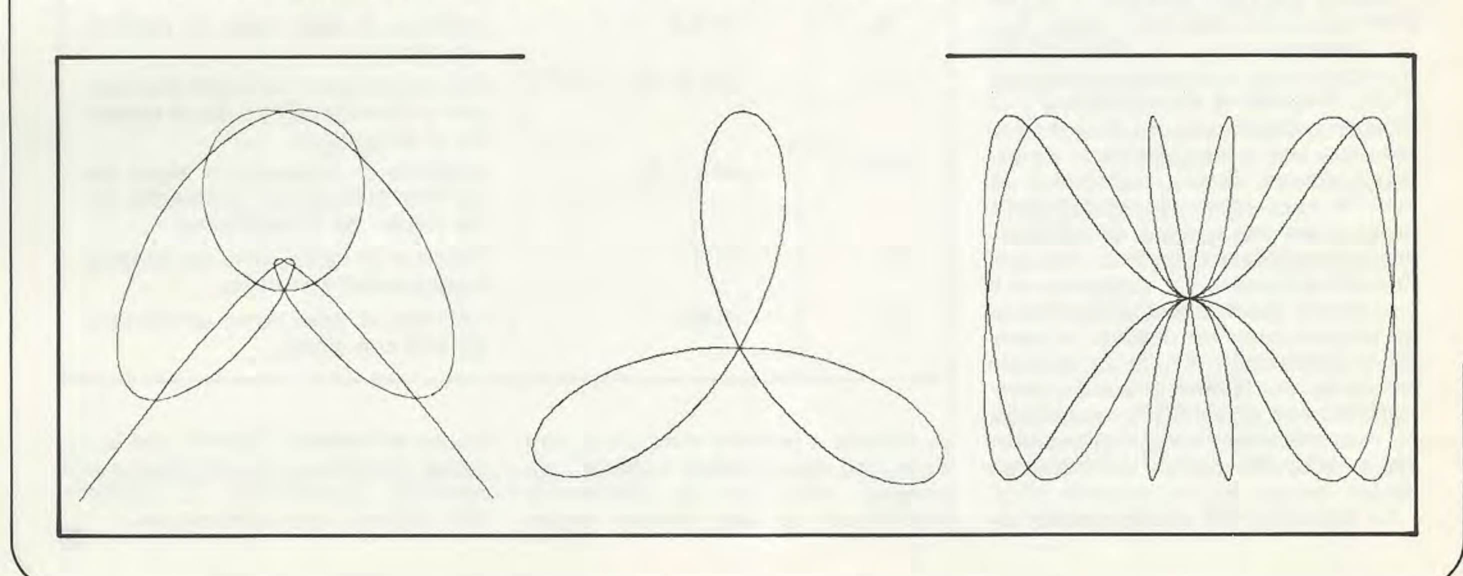
Como el «plotter» requiere coordenadas X, Y y en la línea 90 se obtienen coordenadas polares, es necesario calcular las proyecciones sobre los ejes X, Y. Esto se realiza en las líneas 100 y 110.

Se han introducido las constantes T, A y B para poder obtener gran variedad de dibujos. Con T=3 el incremento angular es de tres grados pero se repite tres veces la exploración. Con A=B=1 se obtiene la forma fundamental de cada valor de T. Dando a A y B valores enteros se obtienen formas muy curiosas que nos pueden recordar insectos o plantas. Con A=B=1 se obtiene el planeador, con (4,5) la hormiga, con (1,6) una mariposa y con (4,4) la flor de loto. La mayoría se han obtenido con T=3. Si el lector desea investigar le recomiendo los siguientes caminos:

— Dar valores no enteros a las variables A y B.

— Manteniendo constante A, variar con incrementos fraccionarios pequeños la variable B hasta el siguiente valor entero.

— Dar a T valores no enteros. Son interesantes los valores de la forma p/q donde p y q son enteros.



— Ejecutar el programa con valores muy grandes de A y B (entre 30000 y 100000).

Volviendo al programa, comentaremos las demás líneas:

LÍNEA 10: se define el radio máximo.
LÍNEA 20: se definen las coordenadas origen del dibujo (en el centro).
LÍNEA 30: el valor T=3 puede variarse.
LÍNEA 40: entrar los valores de A y B.
LÍNEA 50: se abre el canal 1 de dibujo (dirección secundaria 1) y el canal 2 de texto (dirección secundaria 2) hacia el «plotter» (COMMODORE suministra el «plotter 1520» con número de periférico 6.)
LÍNEA 60: se sitúa el bolígrafo en el origen del dibujo.
LÍNEA 70: se declara el punto anterior como nuevo origen de coordenadas.
LÍNEA 80: se inicia el bucle de obtención de valores.

LISTADO 1

```
1 REM PROGRAMA 1
2 REM PARA PLOTTER 1520
3 REM POR ANTONIO GUIX
10 Z=200
20 X0=240:Y0=-240
30 T=3
40 INPUT "A<B";A,B
50 OPEN 1,6,1:OPEN 2,6,0
60 PRINT#1,"M";X0;Y0
70 PRINT#1,"I"
80 FOR AN=0 TO 2*π STEP 2*π/360
90 R=Z*SIN(AN*T)
100 X=R*COS(A*AN)
110 Y=R*SIN(B*AN)
120 PRINT#1,"J";X;Y
130 NEXT AN
140 PRINT#1,"R";-200;-240:PRINT#2,"A="A;"    B="B;"    T="T
150 CLOSE1:CLOSE2:END

READY.
```

LÍNEA 90: se obtiene el módulo R para cada valor angular.
LÍNEAS 100 y 110: se calculan las proyecciones sobre X, Y.
LÍNEA 120: se traza una línea desde el punto anterior al nuevo. (Cuanto menor sea el STEP mayor definición.)
LÍNEA 130: se cierra el bucle.
LÍNEA 140: se lleva el bolígrafo fuera del dibujo y se imprimen los valores de A, B y T.
LÍNEA 150: se cierran los canales abiertos.

Quien desee investigar presentando los dibujos en pantalla puede utilizar el programa número 2 que está preparado para dibujar a puntos en alta resolución. Es algo lento pero sólo pretende poder ahorrar papel cuando se quiera ver el dibujo que se obtiene antes de imprimirlo.

Los valores de Z, X0, Y0 se han adaptado a las características de la pantalla en alta resolución (320×200 puntos). Se ha añadido la constante K para adaptar la escala X a la Y.

Para obtener los puntos en alta resolución se utiliza el MODO BIT MAP de la guía de referencia del programador del COMMODORE 64.

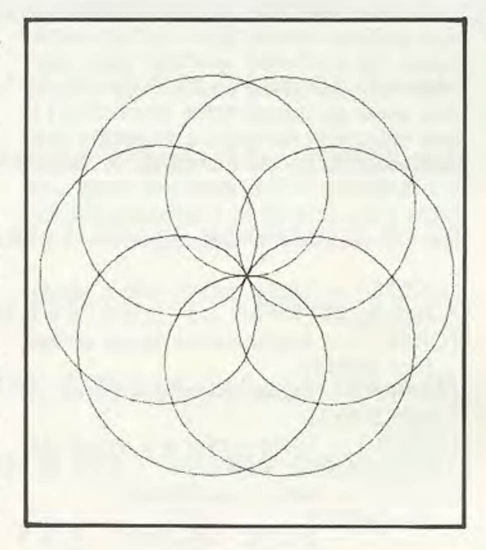
En la línea 130 se define el origen de coordenadas en el centro de la pantalla. La línea 170 presenta pixel a pixel las puntos (x, Y).

Nota: El programa 1 puede ejecutarse en el VIC 20. El programa 2 solamente en el COMMODORE 64. ■

LISTADO 2

```
1 REM PROGRAMA 2
2 REM POR ANTONIO GUIX
10 Z=100
20 X0=160:Y0=100
30 T=3:K=1.25
40 INPUT"A<=B";A,B
50 BA=2*4096:POKE53272,PEEK(53272)OR8:REM SITUAR BIT MAP EN 8192
60 POKE53265,PEEK(53265)OR32:REM ENTRAR EN MODO BIT MAP
70 FORI=BATOBA+7999:POKEI,0:NEXT:REM BORRAR BIT MAP
80 FORI=1024TO2023:POKEI,3:NEXT:REM COLORES CYAN Y NEGRO
90 FOR AN=0 TO 2*π STEP 2*π/360
100 R=Z*SIN(AN*T)
110 X1=INT(R*K*COS(A*AN))
120 Y1=INT(R*SIN(B*AN))
130 X=160+X1:Y=100-Y1
140 RO=INT(Y/8):CH=INT(X/8):LI=YAND7
150 BI=7-(XAND7)
160 BY=BA+RO*320+CH*8+LI
170 POKEBY,PEEK(BY)OR2↑BI
180 NEXTAN
190 GETA$:IFA$=""THEN190
200 POKE53265,PEEK(53265)AND223:REM MODO BIT MAP OFF
210 POKE53272,PEEK(53272)AND247
220 PRINT"s":END

READY.
```



BOLETÍN DE SUSCRIPCIÓN - club commodore

NOMBRE .... EDAD ....
DIRECCIÓN ....
POBLACIÓN .... (..............) PROVINCIA ....
TELÉF. .... MARCA Y MODELO DEL ORDENADOR ....
....
APLICACIONES A LAS QUE PIENSA DESTINAR EL EQUIPO ....
....
....

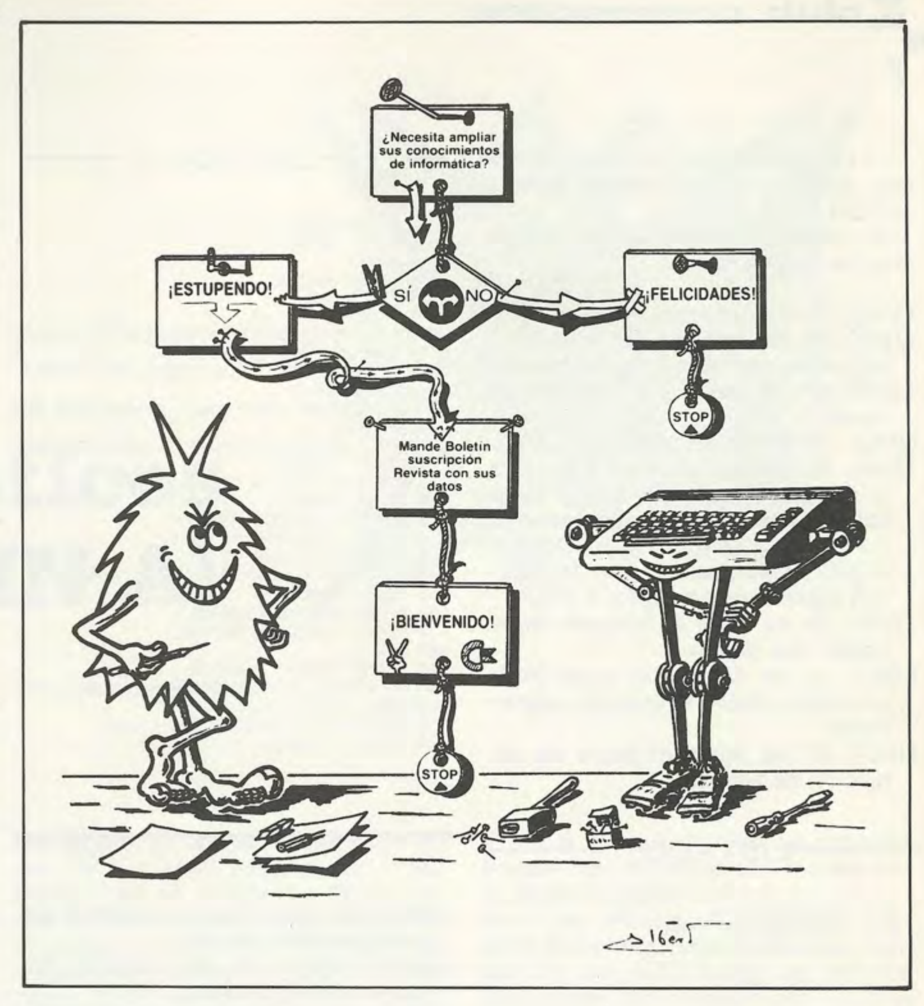
Deseo iniciar la suscripción con el n.° 14

Firma,

(Enviar a la dirección del dorso)

DESEO SUSCRIBIRME A **"CLUB COMMODORE"** POR UN AÑO AL PRECIO DE **1.980** PTAS., QUE PAGARÉ CONTRA REEMBOLSO AL RECIBIR EL NÚMERO CON EL QUE SE INICIA LA SUSCRIPCIÓN. DICHA SUSCRIPCIÓN ME DA DERECHO, NO SÓLO A RECIBIR LA REVISTA (ONCE NÚMEROS ANUALES), SINO A PARTICIPAR EN LAS ACTIVIDADES QUE SE ORGANICEN EN TORNO A ELLA Y QUE PUEDEN SER: COORDINACIÓN DE CURSOS DE BASIC, INTERCAMBIOS DE PROGRAMAS, CONCURSOS, ETC.

clave para interpretar los listados de CLUB COMMODORE



Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD] = Tecla cursor hacia abajo (sin SHIFT)
[CRSRU] = Tecla cursor hacia arriba (con SHIFT)
[CRSRR] = Tecla cursor a la derecha (sin SHIFT)
[CRSRL] = Tecla cursor a la izquierda (con SHIFT)
[HOME] = Tecla CLR/HOME (sin SHIFT)
[CLR] = Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 0.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo — por ejemplo [COMM+] o [SHIFA] —. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente. ■

SERIE 700 (II)

los bancos de memoria y la extensión del BASIC

por **JORDI SASTRE** (MEC. SOFT.)

Continuamos con la serie 700. En esta ocasión hablaremos de dos importantes diferencias del 700 con respecto al 8000 a nivel interno: los bancos de memoria y la extensión de sentencias BASIC.

BANCOS DE MEMORIA

Con el microprocesador 6509 se tiene un direccionamiento de 20 bits: cuatro para el banco y dieciséis para la dirección. En otras palabras, trabaja sobre dieciséis bancos de 65.536 bytes cada uno; total: 1.048.576 bytes **(1 Mbyte).**

En todos los bancos, el byte 0 y el 1 son constantes. El primero indica el banco de ejecución, es decir, el banco donde se halla el código máquina que se está ejecutando, los vectores de interrupción (IRQ, NMI, BRK), las rutinas, la página cero, etc. El byte 1 es el banco de Indirección, indicando el banco sobre el que deben ejecutarse los comandos PEEK y POKE del BASIC, o LDA(xxxx),Y y STA(xxxx),Y del Código máquina.

En la Tabla 1 reflejamos el contenido de cada banco, según la versión del modelo (128K o 256K).

Claramente se ve que hay un banco para texto de programas BASIC; uno o tres (según versión 128K o 256K) para datos; el banco 15 para el siste-

ma; y los demás abiertos para expansión. Precisamente es el banco 15 el que tiene más interés por abarcar todo el sistema operativo, página cero, «buffers», etc. En la Tabla 2 detallamos el contenido del banco 15.

TABLA 2

DIRECCIÓN	CONTENIDO
$0000	Banco de Ejecución 6509.
$0001	Banco de Indirección 6509.
$0002-$03FF	RAM: «Flags» y punteros usados por Kernal, Editor de Pantalla, BASIC y DOS.
$0400-$07FF	RAM no usada.
$0800-$0FFF	RAM usada por el sistema.
$1000-$1FFF	ROM: DOS para los «floppy-disks» integrados.
$2000-$7FFF	24K del «cartridge» (RAM o ROM).
$8000-$BFFF	ROM: Intérprete BASIC 4.
$C000-$CFFF	ROM no usada.
$D000-$D7FF	RAM de la pantalla 80 × 25 (2000 caracteres).
$D800-$D8FF	Controlador de Video Motorola 6845.
$D900-$D9FF	MOS 6523 TPI: Interfaz «floppy-disks» integrados.
$DA00-$DAFF	MOS 6581 SID: Chip de Sonido (Sound Interface Device).
$DB00-$DBFF	MOS 6526 CIA: Interfaz para Z80 o 8088.
$DC00-$DCFF	MOS 6526 CIA: «Complex Interface Adapter».
$DD00-$DDFF	MOS 6551 ACIA: Interfaz Asíncrono RS-232.
$DE00-$DEFF	MOS 6525 TPI: «Tri-Port Parallel Interface # 1.
$DF00-$DFFF	MOS 6525 TPI: Tri-Port Parallel Interface # 2.
$E000-$EFFF	ROM: Editor de Pantalla.
$F000-$FFFF	ROM: Kernal.

EXTENSIÓN DE SENTENCIAS BASIC

El BASIC de la serie 700 está ampliado con una serie de sentencias nuevas, que son:

BANK: Permite cambiar de banco para las sentencias PEEK y POKE.

BLOAD: Efectúa una carga binaria de un fichero programa (grabado con BSAVE) directamente sobre la RAM. Útil principalmente para la carga de rutinas en código máquina. Idéntico al LOAD del monitor.

BSAVE: Grabación binaria de cualquier zona de memoria sobre un fichero programa. Equivale al DSAVE pero aplicado a rutinas en código máquina o posiciones de memoria.

DELETE: Comando para borrar una o varias líneas del programa existente en memoria.

DCLEAR: Inicializa la unidad de discos cerrando todos los canales que hubiera abiertos.

DIRECTORY: Este comando ya existe en el 8000 pero aquí se le ha incorporado un «offset» para listar o visualizar sólo parte del directorio.

DISPOSE: Aborta un bucle FOR... NEXT o GOSUB...RETURN, generalmente por haberse producido algún error (ver TRAP).

ERR$(ER): Devuelve el mensaje de error ER (ver TRAP).

EL , ER: Variables asignadas por TRAP con la línea y el número del error que se ha producido.

IF: Comando al que se le ha incorporado la función ELSE. Es decir, el formato sería así:
IF (comparación) THEN (acción-1) :ELSE (acción-2).

INSTR: Permite conocer la posición que ocupa una «sub-string» dentro de una «string».

KEY: Comando para asignar un contenido a cualquiera de las teclas de función.

PRINT: Otra instrucción que ha sido modificada incorporando la opción USING.

PUDEF: Permite redefinir ciertos símbolos del formato asignado en el PRINT USING.

RESUME: Permite continuar la ejecución del programa en un punto indicado después de producirse y gestionarse un error (ver TRAP).

RESTORE: Se le ha incorporado un parámetro para indicar el número de línea de la instrucción DATA donde desea posicionarse el puntero.

TRAP: Permite bifurcar a la línea indicada cuando se produzca un error durante la ejecución del programa, asignando las variables ER y EL. RESUME devuelve el control después de tratado el error.

Con el ordenador se entrega un completo manual donde se describen detalladamente no sólo estas nuevas sentencias sino todas las que componen el BASIC 4. ■

TABLA 1

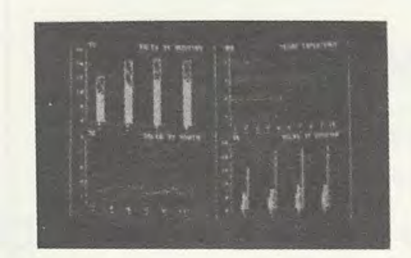
Banco	Uso en 700 128K	Uso en 700 256K
0	(Expansión Futura)	
1	RAM: Texto de programas BASIC	
2	RAM: Tablas	RAM: Tablas, Variables y «Strings»
3	RAM: Variables	(Expansión RAM)
4	RAM: «Strings»	(Expansión RAM)
del 5 al 14	(Expansión RAM de 640K)	
15	ROM y RAM SISTEMA: Kernal, «Screen-Editor», DOS, BASIC, I/O etc.	



SAKATI, S. A. SOFTWARE PROFESIONAL

Ardemans, 24 - Madrid-28 - Tel. 2567794 - Telex 44222 CICI E

¿ES UD. USUARIO DE UN COMMODORE-64...? SI ES ASÍ ENTONCES NUESTROS PROGRAMAS SON PARA UD.



GRAFICOS-64

Le permite trabajar en modo alta resolución. Traza puntos, líneas, rellena áreas de color, define colores de pantalla y fondo, abre ventanas en la pantalla, escribe y superpone textos en los gráficos, invierte textos. 35 nuevos comandos harán de su trabajo una diversión. MANUAL EN CASTELLANO.

P.V.P. en disco 9.000.— Pts.
P.V.P. en cinta 7.500.— Pts.

EXTENSION DE BASIC

Potencie el sistema operativo de su COMMODORE-64 con este cartucho que añade 42 nuevos comandos - de gran utilidad. Trabaje con BASIC 4.0 de COMMODORE. Podrá localizar y corregir programas con gran facilidad, pasar programas de un CBM-8000 a un 64, potenciar el editor, y conectar sin ningún interface, sólo con un cable, su 64 a una impresora CENTRONIC. MANUAL EN CASTELLANO.

P.V.P. 25.000.— Pts.

DESENSAMBLADOR

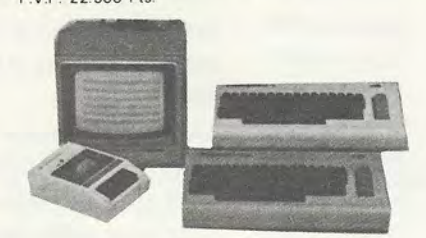
Le permite desensamblar cualquier programa realizado en código máquina. INSTRUCCIONES EN CASTELLANO.

P.V.P. 2.000 Pts. en cinta.

BASE DE DATOS I

Es algo más que un potente generador de aplicaciones. Los programas por aplicación son ilimitados. Aporta un nuevo lenguaje con más de 40 nuevos comandos. Podrá diseñar su propia configuración, con registros de hasta 1000 caracteres y gran facilidad de modificación de los mismos. 127 campos por registro. Contará con 15 ficheros por aplicación ilimitados. Los ficheros de clave y de memoria son así mismo ilimitados. Enlace con EASY SCRIP. Calendario aritmético. Esta Base de Datos es el número uno en su género, algo que Vd. debe tener para su trabajo. MANUAL EN CASTELLANO.

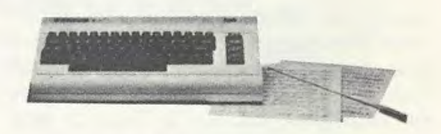
P.V.P. 22.500 Pts.



BASE DE DATOS II

12 opciones presentadas en el menú. Cree nuevos ficheros de datos. Cuenta Vd. con 12 campos. Esta Base de Datos es ideal para personas que empiezan a introducirse en este método de trabajo

P.V.P. 4.000 Pts.



¿LE GUSTARIA COMPONER MUSICA?

Nuestro programa hará de Vd. un melómano. MUSICA-64 controla tres voces en un rango de 8 octavas cada voz. Sonarán en cualquiera de estos 4 instrumentos: órgano, clavicornio, silofón y trompeta. Podrá introducir 2.000 notas por voz, y una vez compuesta su música, archívela en cinta o en disco. INSTRUCCIONES EN CASTELLANO.

P.V.P. en cinta 4.000 Pts.
P.V.P. en disco 5.000 Pts.

QUIMICA

Lecciones asequibles de química para sus hijos y para todos los padres de los hijos que estudian química.

P.V.P. 2.000 Pts. en cinta

VELOS AK-64

Su soporte para programar, capaz de ordenar 500 matrices en 30 segundos.

P.V.P. 2.000 Pts. en cinta

¡NO NOS OLVIDAMOS DE LOS NIÑOS NI DE LA TARDE DEL DOMINGO!

JUEGO DE LOS PALILLOS

Cace los palillos movibles y con joy-stick enciérrelos en el menor área posible, pero ¡ojo! los palillos se mueven en todas direcciones

GUERRA DE TANQUES

Prepárese a rechazar un ataque de la brigada de tanques demostrando su puntería.

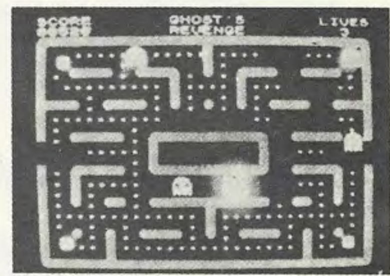
CACTUS

En su desierto particular, tiene Vd. un hermoso cactus al que amenazan los topos, las águilas y los pájaros.

COMECOCOS

El popular juego ahora disponible para su 64.

P.V.P. 2.000 Pts. por juego en cinta
P.V.P. 6.000 Pts. por los 4 juegos en disco



BOLETIN DE PEDIDO

A enviar a: **SAKATI, S. A.** Ardemans, 24. MADRID-28

Ref.		Cdad.	Precio
		TOTAL	

☐ Talon adjunto ☐ Contra Reembolso

Fecha: Firma:

D.

Nº Calle

Código Postal Ciudad

Provincia Teléfono

¡TENEMOS ALGO MUY IMPORTANTE! CONTROL DE ALMACEN

Toda aquella empresa que tenga problemas de almacén no debe pararse en este anuncio. Debe llamarnos y le explicaremos con detalle este programa. El espacio no da para más, pero he aquí un avance:

- 1.500 referencias son controladas por almacén, en unidades, litros, kilos, metros, etc., stock mínimo, y lo propio de estos programas, etiquetas, etc.
  Pero además el programa tiene una opción de anotación para los gratos de su empresa, con análisis de rentabilidad de su negocio, sin pasar por contabilidad. De manejo sencillo. TODO EN CASTELLANO. El programa se presenta en disco.

P.V.P. 35.000 Pts. Ref: SD14

FICHEROS CBM (IV)



técnicas de acceso directo

por **MANUEL AMADO**
(M.E.C.-SOFT.)

B) CANALES Y ÓRDENES PARA EL ACCESO DIRECTO

Como vimos en el artículo anterior, para poder acceder y procesar un determinado sector, tenemos que abrir dos canales de comunicación, uno para comunicarnos entre la CPU y el «buffer» —el canal de acceso directo— y otro para enviar las órdenes o comandos DOS al floppy para la comunicación entre el «buffer» y el sector del disquete deseado. Para abrir estos canales, el formato de la instrucción OPEN en cada caso es el siguiente:

A) Canal de acceso directo:

OPEN NF,NP,SA,"#[NB]"

Donde NF es el número de fichero lógico, NP el número de periférico y SA la dirección secundaria asociada. El símbolo "#" es el que le indica al BASIC y al DOS que vamos a acceder a un «buffer» del floppy, y no a un fichero del disco. [NB] es un parámetro opcional mediante el cual podemos indicar qué «buffer» queremos tratar. Evidentemente, si este «buffer» ya estaba asignado previamente a un fichero u otro canal, el DOS enviará el mensaje de error 70, NO CHANNEL, indicador de que ese canal o «buffer» ya estaba asignado.

B) Canal de órdenes:

OPEN NF,NP,15

Donde NF ha de ser 1 ó 15. Una vez ejecutado este comando, mediante sentencias del formato PRINT#NF, O$, donde O$ es una cadena de caracteres que contiene la orden, puedo hacer que el DOS ejecute la orden o comando.

Comandos del DOS:

A continuación voy a detallar los diferentes comandos DOS necesarios para trabajar en acceso directo. Hay otros comandos, que no intervienen, o si lo hacen es escasamente, en el acceso directo y que, por tanto, los dejo para artículos posteriores. Estos comandos son:

COMANDO		FORMATO
BLOCK-READ	(Lectura bloque)	"B-R";CH;DR;T;S
BLOCK-WRITE	(Escritura bloque)	"B-W";CH;DR;T;S
BLOCK-POINTER	(Puntero bloque)	"B-P";CH;P
BLOCK-ALLOCATE	(Asigna bloque)	"B-A";DR;T;S
BLOCK-FREE	(Libera bloque)	"B-F";DR;T;S
USER-1	(Lectura usuario)	"U1";CH;DR;T;S
USER-2	(Escritura usuar.)	"U2";CH;DR;T;S

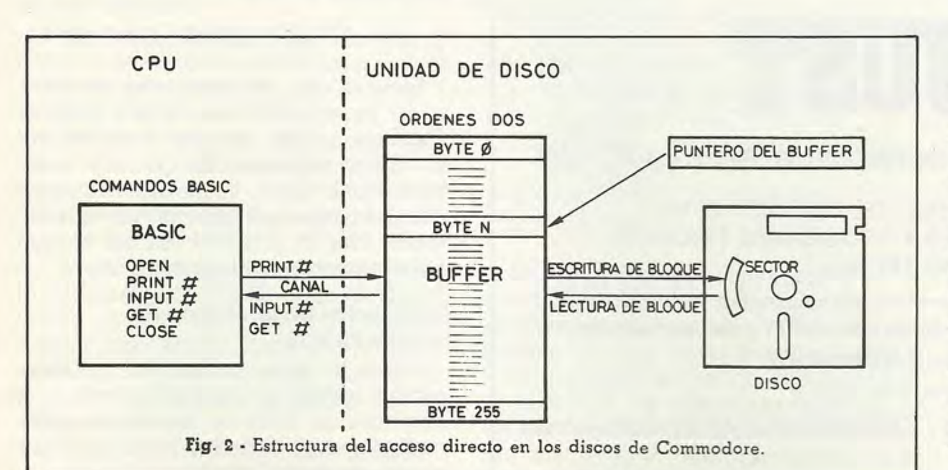


Fig. 2 - Estructura del acceso directo en los discos de Commodore.

Donde:

CH: Número lógico del «buffer» a procesar. Se corresponde con la dirección secundaria, SA, especificada al abrir el canal de acceso directo (OPEN NF,NP,SA,"#").

DR: Número del «drive» en que se halla el bloque que se está procesando. T: Pista donde se halla dicho bloque, y S: Sector de dicho bloque. P: Número del byte del «buffer» en donde nos queremos situar.

Veamos el significado de este misterioso P, o puntero del «buffer». Un «buffer» consta de 256 bytes, numerados del 0 al 255. Ahora bien, los ficheros normales, como pueden ser

(continúa en la pág. siguiente)

técnicas de acceso directo (viene de la pág. anterior)

los secuenciales, relativos, programas, etc., usan sólo 254 bytes para datos, de los 256 bytes útiles que contiene cada sector. Los dos que sobran, los dos primeros (bytes 0 y 1), se los reserva el DOS para punteros, que más tarde veremos detalladamente. Por esta razón, el DOS usa el byte 0 como puntero del «buffer» que se está procesando. Este puntero le indica al DOS cuál es el siguiente byte al último byte accedido, ya sea al ejecutar una sentencia de lectura del «buffer» (GET # o INPUT#) o de escritura (PRINT#). Este byte es el que se modifica al ejecutar un BLOCK POINTER (B-P), y que nos permite acceder directamente a un byte determinado del «buffer».

Vamos a ver primeramente cada comando qué es lo que hace, para luego pasar a estudiar un ejemplo-programa de uso de todas estas órdenes para realizar el acceso directo:

—BLOCK-READ: ("B-R";CH;DR;T;S)

Lee el bloque indicado por DR,T, y S volcando su contenido en el «buffer» referenciado por CH. Lee tantos caracteres del bloque como le indica el puntero del «buffer» en ese momento. No obstante, no se recomienda en absoluto su uso (la razón la explico luego) sustituyéndose por U1.

—BLOCK-WRITE: ("B-W";CH;DR;T;S)

Escribe el contenido del «buffer» indicado por CH en el bloque que señala DR, T y S. Este comando tiene un inconveniente, que se graba la información desde el primer byte del «buffer» hasta el último CHR$(13) que se encuentre. Y además, deja el puntero del «buffer» apuntando a dicho carácter. O sea, que graba hasta dónde se escribió con la orden PRINT# NF en el «buffer». Ello conlleva que al ejecutar un BLOCK-READ, éste sólo leerá hasta donde le indica el puntero, que se ha grabado en el bloque al ejecutar el B-W, y no nos dejará acceder hasta el final del «buffer», perdiendo consecuentemente el resto de la información. Esto queda solucionado utilizando, en lugar de B-R y B-W, U1 y U2 respectivamente. Ver los comentarios referentes a U1 y U2.

—BLOCK-POINTER: ("B-P";CH;P)

Coloca el puntero del «buffer» indicado por CH en el byte P. Es un concepto análogo a la instrucción RECORD # NF,(NR),(B) que permite situarnos directamente en el byte NB del registro NR del relativo NF.

— BLOCK-ALLOCATE: ("B-A";DR;T;S)

Da de alta en el BAM (pone a 0 el bit correspondiente), el bloque indicado por DR,T y S. Ello es útil en el momento de crear un fichero de acceso directo (A.D.), dado que la creación, desde un punto de vista simplificado, se reduce a la reserva del espacio en disco correspondiente a dicho fichero. Esto se consigue ejecutando un bucle de B-A, con lo que reservamos en el BAM los bloques necesarios. Una vez un bloque figura como ocupado, asignado, en el BAM, el DOS lo omitirá en cualquier operación posterior en que se grabe un fichero o se escriba o creen ficheros secuenciales o relativos. Si el bloque ya estaba asignado previamente, el DOS enviará un error de NO BLOCK (bloque no disponible), indicándonos al mismo tiempo el siguiente bloque libre.

—BLOCK-FREE: ("B-F";DR;T;S)

Da de baja, libera, del BAM el bloque indicado por DR, T, y S. Con ello, el bloque está disponible para que el DOS lo asigne si es necesario a un fichero o programa.

—USER-1: ("U1";CH;DR;T;S)

Ejecuta una lectura del bloque indicado, volcándolo en el «buffer» CH. Como además fuerza el puntero del bloque a 255, volcará el bloque entero desde el disco al «buffer». Ello nos permite poder acceder a todos los bytes del bloque leído. Es la orden que se usará cuando se desee volcar el contenido del bloque deseado en el «buffer» seleccionado, sustituyendo totalmente a "B-R".

—USER-2: ("U2";CH;DR;T;S)

Ejecuta una escritura del «buffer» en el bloque indicado. Graba todo el «buffer», desde el byte 1 al 255, en el bloque indicado. Se ignora el puntero de «buffer». Es la orden que se usará cuando se desee grabar el contenido de un «buffer» en un bloque determinado, sustituyendo a "B-W".

Realización de un fichero de acceso directo:

Antes de pasar a describir las diferentes rutinas de creación, lectura y escritura de datos en un determinado registro, consideremos los siguientes aspectos preliminares:

—El fichero vendrá definido por una serie de parámetros, que pueden ser:

La longitud máxima de un registro será 254.

NR: Número de registros del fichero.

LR: Longitud del registro.

OF: Offset. Es el número de bloque en el que comienza el fichero. Este número se calcula de la siguiente forma: si el primer registro de nuestro fichero empieza en la pista T y el sector S, y hay NS sectores por pista en el rango de pistas considerado (ver artículo anterior), tendré que OF = T*NS + S

PT = Número de registros por bloque. PT = INT(254/LR).

Consecuentemente, el número de registro de un fichero de acceso directo está íntimamente relacionado con los parámetros de acceso directo T, S y P (Pista, Sector y Puntero), y a cada número de registro del fichero, le corresponderá una única combinación de T, S y P. La rutina de decodificación del número de registro RL en su correspondiente T, S y P será la siguiente, teniendo en cuenta que es sólo válida para una porción del disquete con el mismo rango de pistas (o sea, en la que todas las pistas consideradas tienen el mismo número de sectores por pista (NS)):

```
10 T=INT((OF+(INT((RL—1)/PT)))/
   NS) : S=OF+INT((RL—1)/PT)—
   T*NS
20 P=INT(254/PT*(RL—1—INT((RL—
   1)/PT)*PT))+1 : RETURN
```

Vuelvo a subrayar la importancia de esta rutina de decodificación, pues nos permite acceder a un registro por su número absoluto de registro, y que en realidad en el disquete estará en la pista T, sector S y a partir del byte P del bloque considerado.

a) Creación de un fichero de acceso directo:

Primeramente debo destacar que el fichero de acceso aleatorio no va a tener ninguna entrada de directorio y, por lo tanto, la creación se va a limitar a la reserva del espacio en disco que necesita el fichero. Por lo tanto, en un disco en el que hayan ficheros de acceso aleatorio, NO SE PUEDE EJECUTAR un COLLECT, debido a que ello provocaría que todos los bloques del fichero de A.D. se darían de baja en el BAM y pasarían a estar disponibles, dado que no tiene entrada de directorio. La rutina de creación del fichero (reserva de los bloques), sería:

```
100 REM RESERVA DEL ESPACIO EN
    DISCO PARA EL FICHERO
105 REM DE ACCESO DIRECTO. DR:
    DRIVE DEL FICHERO
110 REM ABERTURA DE LOS CANA-
    LES DE ACCESO DIRECTO
115 OPEN 2,8,4,"# " : OPEN 15,8,15
120 FORI=1TONR:GOSUB10:PRINT#
    15,"B-A";DR;T;S:NEXT
125 REM CREA UN FICHERO
    SECUENCIAL PARA FORZAR QUE
130 REM EL NUEVO BAM GENERADO
    CON "B-A" SE GRABE
135 REM EN EL DISQUETE (DOS > 2).
140 REM PARA DOS<2 HAY QUE
    INICIALIZAR EL DISQUETE.
145 DOPEN#3,"@PEPE",W,DR:
    PRINT#3,"A":DCLOSE#3:
    DCLOSE
150 SCRATCH"PEPE",DR:END
```

Para las siguientes subrutinas, se presupone que antes de ser llamadas ya han sido abiertos previamente el canal de acceso directo (p. ejm. OPEN 2,8,4,"# "), y el de órdenes (p. ejem. OPEN 15,8,15).

b) Lectura de un registro determinado:

```
200 REM LECTURA DE UN REGISTRO
    RL DETERMINADO EN A.D.
205 GOSUB10 : B$="": PRINT#15,
    "U1";4;DR;T;S: PRINT#15,
    "B-P";CH;P : FORI=1TOLR :
    GET#2,A$
210 B$=B$+LEFT$
    (A$+CHR$(0),1) : NEX : RETURN
```

El contenido del registro se devuelve en la variable B$. La lectura se efectúa por GET # y no por INPUT # por si hubiesen campos empaquetados que no se pudieran leer mediante INPUT#. No obstante, si los campos no están empaquetados, se podría leer el registro directamente mediante un INPUT#, siempre y cuando la longitud del dato a leer no supera-se los 80 caracteres.

c) Grabación de un registro determinado:

```
300 REM RUTINA DE GRABACIÓN
    DE UN REGISTRO RL
305 REM EL CONTENIDO DEL
    REGISTRO SE HALLA EN R$
310 GOSUB10 : PRINT#15,"U1";
    4;DR;T;S : PRINT#15,"B-P";4;P :
    PRINT#2,R$;
315 PRINT#15,"U2";4;DR;T;S :
    RETURN
```

En fin, ya podéis pues empezar a montaros vuestros maravillosos programas para la gestión de ficheros de acceso directo. En el próximo artículo empezaré a hablar de la estructura en disco de los diferentes ficheros, y viajar de una vez a las profundidades del disquete. Y mientras tanto, ¡que vosotros lo accedáis bien! ■

MARKETCLUB

- **CBM 4.032. Intercambio programas. José Marcé Mestres. Calle Sevilla, 5. Tel. (93) 803 77 51, de 8 a 3. VILANOVA DEL CAMÍ (Barcelona).**
- **ACCESORIOS VIC-20: Ampliación de memoria 16K más varios programas, 13.000. Módulo de expansión para 6 cartuchos, 10.000. Cartucho lenguaje FORTH y manual, 7.000. Las tres cosas sólo por 25.000. Jaime. Tel. 245 46 56. BARCELONA.**
- **En Barcelona, clases de informática. PLAZAS LIMITADAS. Lenguaje BASIC. Prácticas con micro-ordenador VIC-20. Prof. E. Martínez de Carvajal. Información: Tel. (93) 345 10 00. Señorita María José (mañanas) o (93) 345 87 75. Sr. Martínez (fuera de horas de oficina).**
- **Desearía vender por 40.000 ptas.: VIC-20, con cartucho Super Expander, las dos partes del Curso, el «joy-stick» y un juego Indescomp, todo comprado en diciembre del 82. Fernando Martínez, calle La Roda, 39, 5º D. Teléfono 23 41 82. ALBACETE.**
- **Tengo un PET 2001/8K y desearía tener el cassette «Monitor para lenguaje Máquina». También desearía contactar con usuarios o clubs del PET si los hay. José Manuel Cámara Mas, calle Castor, 32, bloque II, 3º, puerta 1. ALICANTE.**
- **Programación de ordenadores personales; organización explotación de ficheros; programas ordenadores auxiliares, para cuestiones empresariales, profesionales, administrativas, científicas. Mora Mas. Carlos III, 41. Teléfono 339 98 29. BARCELONA-28.**
- **Vendo aplicación de facturación con control de representantes, 9 listados, 6 ficheros, estadísticas, etc. Permite copias de seguridad. Configuración: VIC 8 K, disco e impresora, 40.000 pesetas. Escribir a Jaime Ameller Pons. General Mola, 15, 1º B. CALATAYUD (Zaragoza).**
- **Se ofrece 3008 + C2N de segunda mano en buenas condiciones; precio a convenir. Razón: Domingo Garrofé Trabal. Aragón, 386, 1º 1ª Tel. 211 54 40. BARCELONA-9.**
- **Vendo interfaz y programa para RTTY y CW para el PET a 15 K. Rafael, EA3CGK. Av. Barcelona, 21, A, 4º 2ª. IGUALADA (Barcelona).**

EJEMPLARES ATRASADOS DE «CLUB COMMODORE»

Para poder satisfacer la creciente demanda de números atrasados de nuestra Revista, agotada en todas sus ediciones, hemos puesto en marcha un Servicio para suministrar fotocopias de los ejemplares que nos sean solicitados. Para recibir las fotocopias de una o de varias ediciones, no hay más que enviarnos el boletín con los datos indicados.

SERVICIO DE FOTOCOPIAS

NÚMERO DE LA EDICIÓN SOLICITADA

0	1	2	3	4	5	6	7	8	9	10	11	12	13

(Poner una X debajo del número de edición pedido)

Peticionario: D. ....

Calle .... n.° ....

Población .... D.P. ....

Provincia ....

Forma de envío: contra-reembolso Precio de la edición fotocopiada: 250 ptas.

HOJA DE PROGRAMACIÓN - BASIC

Nombre Programa: ______________________ Nº Hoja ____
Nombre Fichero: ______________________ Fecha ___/___/___

LINEA	CONTENIDO LINEA (10, 20, 30)	NOTAS



Microprocesador: 6502 de MOS TECHNOLOGY de 8 bits.
Memoria: 5 Kbytes de RAM ampliables a 32 K. 20 Kbytes de ROM ampliables a 28 K.
Pantalla: 23 lineas de 22 caracteres. Modulador para conectar a un televisor normal. Salida monitor video.
Colores: 8 para el marco, 16 para el fonde de la pantalla y 8 para los caracteres individuales, video inverso.
Gráficos: Semi-gráficos por teclado y alta resolución por redefinición del generador de caracteres (situándolo en RAM). Definición de 176 por 184 puntos.
Teclado: Tipo QWERTY de 62 teclas más cuatro de función definibles por el usuario.
Sonido: Tres voces de tres octavas cada una decaladas una octava entre sí, resultando una extensión total de cinco octavas. Un generador de ruido aleatorio afinable para efectos especiales, un control general de volumen.
Programación: Lenguaje BASIC, intérprete residente en ROM de 8K. Posibilidad de interceptar las funciones del Basic para crear nuevas instrucciones "a medida". El Basic del Vic es uno de los rápidos actualmente en el mercado.
Complementos: Port de usuario de 8 bits entrada/salida más dos señales de sincronismo.
Bus de expansión para ampliaciones de memoria y periféricos.
Port de juegos con conexión para dos potenciómetros (paddles), y una palanca de juegos (joystick).
Almacenamiento de masa: Unidad de cassette C2N de diseño especial para registrar programas y datos.
Ampliación de memoria: En caso de ser necesario conectar más de un cartucho al mismo tiempo, está disponible un módulo (VIC 1020) que permite la conexión simultánea de hasta seis cartuchos.

VIC-1541 UNIDAD DE DISCO

Capacidad total: 174848 bytes por disco.
Secuencial: 168656 bytes por disco.
Entradas de directorio: 144 por disco.
Sectores por pista: De 17 a 21.
Bytes por sector: 256.
Pistas: 35.
Bloques: 683 (644 bloques libres).
Soportes de información: Discos standar de 5 1/4 pulgadas, de una sola cara y densidad simple.
Sistema operativo: DOS de COMMODORE inteligente (tiene procesador propio y no ocupa memoria del ordenador central).

VIC-1525 IMPRESORA

Método de impresión: Matriz de 5 × 7 puntos, impacto por un solo martillo.
Modo caracteres: Mayúsculas y minúsculas, símbolos, números y caracteres gráficos del VIC-20.
Modo gráfico: Puntos direccionables (bit image). Siete puntos verticales por columna, 480 columna máximo.
Velocidad: 30 caracteres/segundo, de izquierda a derecha, unidireccional.
Caracteres/Línea: Máximo 80. (Posibilidad de impresión en doble ancho).
Espaciado entre líneas: 6 lineas/pulgada —modo caracteres, 9 lineas/pulgadas— modo gráfico.
Alimentación de papel: Arrastre por tractor.
Ancho de papel: Entre 4,5 y 10 pulgadas.
Copias: Original más dos copias.

CARTUCHOS

Ayuda programador: Facilita la edición y depuración de programas en Basic. Instrucciones y comandos: RENUMBER, MERGE, FIND, CHANGE, DELETE, AUTO, TRACE, STEP, OFF, KEY, EDIT, PROG, DUMP, HELP y KILL.

Super expander: Intercepta el Basic del VIC permitiendo incrementar sus instrucciones y comandos en aplicaciones gráficas de sonido y juegos. Instrucciones y comandos: KEY, GRAPHIC, COLOR, POINT, REGION, DRAW, CIRCLE, PAINT, CHAR, SCNCLR, SOUND, RGE, RCOLR, RDOT, RPOT, RPEN, RJOY y RSND.
Monitor de lenguaje máquina: Facilita enormemente la depuración de programas en lenguaje máquina, es ideal como complemento del Basic para redactar y poner en marcha rutinas de alta velocidad y manejo de datos en tiempo real. Instrucciones y comandos: ASSEMBLE, BREAKPOINT, DISASSEMBLE, ENABLE, VIRTUAL ZERO PAGE, FILL MEMORY, GO, HUNT, INTERPRET, JUMP TO SUBROUTINE, LOAD, MEMORY, NUMBER, QUICK TRACE, REGISTERS, REMOVE BREACPOINTS, SAVE, TRANSFER, WALK y EXIT TO BASIC.
Además existen cartuchos de ampliación de memoria de 3,8 y 16 Kbytes.

CURSO DE INTRODUCCION AL BASIC PARTE I y II.

En forma de libro se ha editado la primera y segunda parte de un curso de Basic que parte "de cero" y está basado en el VIC-20. Van acompañados de dos cassettes con programas y ejercicios para autocontrol.

PLOTTER VIC 1520

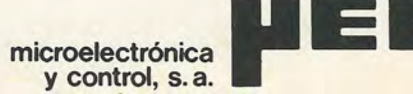
Método de impresión: Dibujo mediante bolígrafos de diseño especial.
Color: 4 colores; negro, azul, verde y rojo con cambio desde programa.
Cabezal: Ploter X-Y tipo tambor.
Velocidad de impresión: Media de 14 car./seg.
Caracteres por línea: Máximo 80 carac., formatos de 80, 40, 20 y 10 carac./línea.
Juego de caracteres: 96.
Velocidad de dibujo: 264 pasos/seg.
Longitud del paso: 0,2 mm. en dirección X e Y.
Velocidad de dibujo de línea: 52,8 mm./seg. en dirección X e Y. 73 mm./seg. en una línea a 45 grados.
Area de dibujo: 480 pasos (96 mm.) en dirección X. Programable en dirección Y (Máx. + — 999 de una sola vez).
Papel: Rollo de 4,5 pulgadas (114 mm.).

MONITOR EN COLOR C-1701

Pantalla: 13 pulgadas (330 mm.).
Capacidad de representación: 25 lineas de 40 caracteres.
Resolución: 320 líneas horizontales.
Compatibilidad: VIC-20 y COMMODORE 64.
Conectable a un registrador de video.
Amplificador y altavoz: Incorporados.







Taquigrafo Serra, 7 5.° Telf. 250 51 03. BARCELONA-29
Princesa, 47 3.° G. Telf. 248 95 70. MADRID-8